

Virgin Tape Transfer 1.0

```
;-----  
;Virgin Tape Transfer 1.0  
;  
;(C) 2005 Nostalgia  
;  
;By Fungus / Nostalgia  
;-----  
  
    *= $0801  
  
    .byte $08,$0b,$d5,$07,$9e  
    .byte $32,$30,$36,$31,$00  
    .byte $00,$00  
  
cd      = $03  
  
;-----  
;header composition  
;  
;f7 = file number  
;f8 = d011 control  
;f9 = load address lo  
;fa = load address hi  
;fb = end address lo  
;fc = end address hi  
;fd = jsr/jmp address lo  
;fe = jsr/jmp address hi  
  
buffer  = $0800  
  
    lda $ba  
    sta cd          ;save current  
                   ;disk drive  
  
    lda #$0b        ;setup  
    sta $d011  
    lda #$c1        ;no nmi  
    sta $0318  
    lda #$80  
    sta $0291       ;no char shift  
    lda #$17  
    sta $d018       ;this set  
    lda #$00  
    sta $9d         ;no msgs  
    sta $d020  
    sta $d021  
    lda #$2f        ;i/o config  
    sta $00         ;bit 4 = input
```

```

        lda $01
        and #$07      ;mem config
        ora #$20      ;motor off
        sta $01
        lda #$00
        sta $0289    ;no keys
        sta $0286

        lda #$93
        jsr $ffd2    ;clr screen

        ldx #$00      ;print msg 1
pm1     lda $0b40,x
        sta $0400,x
        lda #$05
        sta $d800,x
        inx
        cpx #$a0
        bne pm1

        lda #$1b      ;screen on
        sta $d011

wplay   lda $01      ;wait for play
        and #$10
        bne wplay

xl      ldx #$00
        lda loader1,x
        sta $04a0,x
        lda loader1+$0100,x
        sta $05a0,x
        lda loader1+$0200,x
        sta $06a0,x
        inx
        bne xl

        jmp $04a0     ;begin real loader

;-----
;virgin tape routine

loader1 = *
        *= $04a0
        .offs loader1-*

        cli
        jsr loadhead
        jsr namefile
moreload
        jsr putindex

```

```

    jsr printload
    jsr printname
    jsr load
    jsr save
    jmp moreload
load
    sei

    lda #$7f
    sta $dd0d
    sta $dc0d
    lda $dd0d
    lda $dc0d

    lda #$00
    sta $c0      ;in byte
    sta $02      ;chksum

                    ;set load buffer
    lda #<buffer
    sta $ae
    lda #>buffer
    sta $af

    lda #$05      ;motor on/roms out
    sta $01

    lda #$5e      ;015e threshold
    sta $dd04
    lda #$01
    sta $dd05
    lda #$19      ;cnt cycles, 1 shot
    sta $dd0e

resync  lda #$40
        sta $c1      ;pilot countdown
        bne wsync2
wsync1  jsr getbit
wsync2  jsr getbit      ;pilot
        cmp #$aa
        bne resync
        dec $c1
        bne wsync1

wsync3  jsr getbit      ;sync
        jsr getbit
        lsr a
        bcs resync
        cmp #$50
        bne wsync3

```

```
readhdr    ldy #$00      ;get header
           jsr getbyte
           sta $f7,y  ;header
           sta $07,y  ;header copy
           iny
           cpy #$08
           bne readhdr
           ldy #$00

loadloop   jsr getbyte ;load loop
           dec $01
           sta ($ae),y
           inc $01
           eor $02
           sta $02

           inc $ae      ;real load addy
           bne skip1
           inc $af

skip1      inc $f9      ;emulated load addy
           bne skip2
           inc $d020    ;load effect 2
           inc $fa

skip2      lda $fa
           cmp $fc
           bne loadloop
           lda $f9
           cmp $fb
           bne loadloop

           jsr getbyte ;check checksum
           cmp $02
           beq loadok
           jmp loaderr

loadok     lda #$00      ;timer off
           sta $dd0e
           lda #$37      ;motor off/roms in
           sta $01
           cli
           rts

getbyte    lda #$01      ;fetch a byte
           sta $c0

bitget     jsr getbit
           bcc bitget
           rts

getbit     lda #$10
```

```
wbit    bit $dc0d    ;wait flag line
        beq wbit
        lsr $dd0d    ;get bit
        rol $c0
        lda #$19
        sta $dd0e    ;restart countdown
        lda $d020    ;load effect 1
        eor #$05
        sta $d020
        lda $c0      ;get current byte
        rts
```

```
;-----
;load header
```

```
loadhead
```

```
    lda #$01        ;open tape file
    tax
    tay
    jsr $ffba
    lda #$00
    jsr $ffbd
    lda #$00
    sta $c0
    sta $93
    jsr $f7d7        ;load header/file
    jsr $f84f
    lda $01          ;motor off
    ora #$20
    sta $01
    rts
```

```
;-----
;save routine
```

```
save
```

```
    lda #$00        ;save file
    sta $d020        ;border = black
    sta resave+$01
    lda #$1b
    sta $d011        ;screen on
    lda #<buffer
    sta $fb          ;init save addy
    lda #>buffer
    sta $fc
    jsr printsave

    lda #$01        ;open file for
    ldx cd           ;write
    ldy #$01
    jsr $ffba        ;setlfs
```

```
        lda #$10
        ldx #<filename
        ldy #>filename
        jsr $ffbd    ;setname
        jsr $ffc0    ;open
        ldx #$01
        jsr $ffc9    ;chkout
        lda $09      ;start addy low
        jsr $ffa8    ;send
        lda $0a      ;start addy high
        jsr $ffa8    ;send

saveb   ldy #$00      ;save the file
        sei
        inc $01
        lda ($fb),y
        dec $01
        jsr $ffa8
        inc $d020
        dec $d020

        inc $fb
        bne asave
        inc $fc

asave   lda $fc
        cmp $af
        bne saveb
        lda $fb
        cmp $ae
        bne saveb

        lda #$01
        jsr $ffc3    ;close the file

        lda #$01
        jsr $ffc3    ;close the file
        jsr chkerr

resave  lda #$00
        bne save
        rts

chkerr  lda #$00
        jsr $ffbd
        lda #$0f
        tay
        ldx cd
        jsr $ffba
        jsr $ffc0
```

```

        lda cd
        jsr $ffb4    ;talk
        lda #$6f
        jsr $ff96    ;tlksa
        jsr $ffa5
        cmp #$30
        beq noerr
        inc resave+$01
noerr   jsr $ffa5
        cmp #$0d
        bne noerr
        jsr $ffab    ;untlk

        lda #$0f
        jsr $ffc3
        lda resave+$01
        bne waitsave
        rts

waitsave
        ldx #$07
fl      lda ft,x
        and #$3f
        sta $0478,x
        dex
        bpl fl
loopsave lda $d020
        eor #$06
        sta $d020
        lda #$ef
        cmp $dc01
        bne loopsave
        lda #$00
        sta $d020
        rts

;-----
;utility subroutines

loaderr
        ;print tape load
        ;error msg

perr1   ldx #$17
        lda et1,x
        and #$3f
        sta $0478,x
        dex
        bpl perr1
        lda $d020
endloop eor #$02

```

```
        sta $d020
        jmp endloop ;endless loop

printload
        ldx #$07      ;print loading
ll      lda lt,x      ;text
        and #$3f
        sta $0478,x
        dex
        bpl ll
        rts

printsave
        ldx #$07
dl      lda st,x      ;print save text
        and #$3f
        sta $0478,x
        dex
        bpl dl
        rts

printname
        ldx #$0f      ;print name
ln      lda filename,x
        and #$7f
        sta filename,x
        and #$3f
        sta $0480,x
        dex
        bpl ln
        rts

putindex
        lda fileidx
        inc fileidx
        pha
        lsr a
        lsr a
        lsr a
        lsr a
        jsr convert1
        sta filename+$0e
        pla
        and #$0f
        jsr convert1
        sta filename+$0f
        rts

convert1 ora #$30
        cmp #$3a
        bcc convert2
```



```

        adc #$06
convert2 rts

namefile
        lda #$20
        ldx #$0f
clrname sta filename,x
        dex
        bpl clrname
namer   ldy #$00      ;name file
        ldx #$00
dname   lda $0341,y
        beq skip
        cmp #$20
        beq putname
        cmp #$41      ;less than A
        bcc check2
        cmp #$5b      ;more than Z
        bcs skip
        bcc putname
check2  cmp #$30      ;less than 0
        bcc skip
        cmp #$3a      ;more than 9
        bcs skip
putname sta filename,x
        inx
skip    iny
        cpy #$10
        bne dname
        rts

```

;-----

;Transfer Messages

fileidx .byte \$00

et1 .text "Load Error! "

.text " "

ft .text "DiskFull"

lt .text "Loading "

st .text "Saving "

filename .text " "

From: <https://codebase64.org/> - Codebase 64 wiki

Permanent link: https://codebase64.org/doku.php?id=base:virgin_tape_transfer_1.0

Last update: 2015-04-17 04:34



